



Appeal
Brief
5-30-02
K. P. Muller
Patent

Attorney's Docket No. P2380-505

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)
David G. OPSTAD et al) Group Art Unit: 2672
Application No.: 09/306,888) Examiner: T. Havan
Filed: May 7, 1999) Appeal No.
For: AUTOMATIC SYNTHESIS OF)
FONT TABLES FOR CHARACTER)
LAYOUT)

RECEIVED
MAY 24 2002
Technology Center 2600

BRIEF FOR APPELLANT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

This appeal is from the decision of the Primary Examiner dated October 23, 2001 (Paper No. 7), finally rejecting claims 1-31, which are reproduced as an Appendix to this brief.

A check covering the [] \$160.00 (220) [X] \$320.00 (120) Government fee and two extra copies of this brief are being filed herewith.

The Commissioner is hereby authorized to charge any appropriate fees under 37 C.F.R. §§1.16, 1.17, and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 02-4800. This paper is submitted in triplicate.

TABLE OF CONTENTS

	<u>PAGE</u>
I. <u>Real Party in Interest</u>	-1-
II. <u>Related Appeals and Interferences</u>	-1-
III. <u>Status of Claims</u>	-1-
IV. <u>Status of Amendments</u>	-1-
V. <u>Summary of the Invention</u>	-1-
VI. <u>The Issues</u>	-3-
VII. <u>Grouping of Claims</u>	-3-
VIII. <u>Argument</u>	-4-
A. <u>The Brock Patent is Not Concerned With the Problems Addressed By the Present Invention</u>	-4-
B. <u>The Rejection Does Not Establish a <i>Prima Facie</i> Case of Obviousness</u>	-5-
1. <u>Claims 1, 11, 19 and 26</u>	-6-
2. <u>Claims 6, 16, 22 and 29</u>	-8-
3. <u>Claims 7 and 8</u>	-9-
4. <u>Claims 10, 15, 21 and 28</u>	-9-
5. <u>Claims 4, 5, 13, 14, 20 and 27</u>	-10-
IX. <u>Conclusion</u>	-11-



Application No. 09/306,888
Attorney's Docket No. P2380-505

Real Party in Interest

The application involved in this appeal, and the invention to which it is directed, are assigned in their entirety to Apple Computer Inc., of Cupertino, California.

II. Related Appeals and Interferences

There are no other known appeals or interferences that will directly affect, or be affected by, or have a bearing on the Board's decision in this appeal.

III. Status of Claims

The present application contains claims 1-31, all of which are pending and stand finally rejected.

IV. Status of Amendments

There were no amendments filed subsequent to the final Office Action.

V. Summary of the Invention

The claimed invention is directed to fonts that are employed in computer systems to display and/or print images of characters. In essence, a font comprises a number of data tables that relate to various parameters and other features that determine the appearance and "behavior" of the characters in a font as they are being displayed or printed. For example, one table may contain the various images, or glyphs, that are associated with each character. Other tables may contain data such as kerning, i.e., the spacing between characters, the metrics or dimensions of glyphs, variable properties such as line widths, and the like. The information provided by these tables is employed for a number of different purposes. For instance, the data in some of the tables is employed when a line of characters is to be laid out for display or printing purposes. (Page 1, line 24 to page 2, line 5.)

With reference to Figure 2, when a user types a character on a computer's keyboard 24, a call is made to the computer's imaging system 38 to display the character

corresponding to the keystroke. The process that is undertaken to display the character is described in the application at page 6, line 14 to page 8, line 15. As a string of characters are supplied to the imaging system, it may call upon a line layout processor. This processor adjusts the position of individual glyphs relative to one another, and performs further modifications of the glyphs to assemble a complete line of characters. An example of the general procedures that are performed in the line layout processor is illustrated in Figure 4. As depicted therein, each step employs data contained in the tables of the font to perform the necessary modification of the glyphs. (Page 8, line 16 to page 10, line 20.)

The present invention is particularly concerned with the possibility that a given font may not contain all of the tables necessary for the imaging system, e.g. the layout processor, to perform all of the operations on the glyphs. For instance, an older font that was created before the capabilities of the layout processor were developed may not contain all of the requisite tables. In accordance with the invention, tables that are necessary in order to perform a certain operation on a font are automatically synthesized if they are missing from the original definition of the font. Figure 7 schematically depicts a process by which the table is automatically synthesized. The synthesizer employs data contained in some of the tables of a font, such as a glyph mapping table 47, to construct a font map 81. This font map provides certain characterizing information about each glyph in the font. For instance, in the example of Figure 7, the data for each glyph includes a glyph number, an identification value, properties and a unicode value. A mapping table 84 within the synthesizer defines relationships between characters. In a simplistic example depicted in Figure 7, the mapping table identifies the relationship of uppercase and lowercase characters. Based upon the information in this mapping table and the font map 81, the synthesizer creates a new table 86. Each entry in this table maps specific glyphs in the font to one another, in accordance with their uppercase/lowercase relationship. (Page 14, line 5 to page 16, line 11.)

Preferably, once a table is synthesized for a font, it is stored in a persistent manner so that its data is available for subsequent uses of the font. In accordance with another feature of the invention, the synthesized table is not stored as part of the original font

definition. For instance, as illustrated in Figure 5, the original data for a font is contained within a suitcase 46. Automatically synthesized tables are stored in a separate annex 50 that is associated with the font. This annex is a file that contains additional information about the font, but does not affect the original font definition stored in the suitcase. (Page 12, line 15 to page 13, line 9.)

As a result of the capabilities provided by the present invention, it becomes possible for older fonts, or "lightweight" fonts having a limited number of data tables, to be employed in the context of newer display technologies that perform more complex manipulation of the character images. The basic operation of the invention within this context is depicted in the flow chart of Figure 6. When a particular line layout procedure is to be performed, a request for the tables that are pertinent to that procedure is made at step 60. A determination is then made at step 62 whether the requisite tables are present. If so, the procedure is carried out at step 64. If a table is not present, a determination is made at step 70 whether an annex exists. If so, it is checked for the presence of the table. If the table does not exist in the annex, it is automatically synthesized in accordance with the invention, and then placed in the annex. The synthesized table is then used by the procedure to carry out the necessary operations on the glyphs of the font. (Page 13, line 10 to page 14, line 4.)

VI. The Issues

The final Office Action presents a single issue for review on this appeal, namely whether claims 1-31 are unpatentable under 35 U.S.C. § 103 in view of the *Brock et al.* patent (U.S. Patent No. 5,664,086).

VII. Grouping of Claims

Although all pending claims have been grouped in a single ground of rejection, not all of the claims stand or fall together. Rather, various ones of the claims present separate issues of patentability that must be considered independently of other claims. The separate

bases of patentability, and the claims corresponding thereto, are presented in the arguments that follow.

VIII. Argument

A. The Brock Patent is Not Concerned With the Problems Addressed By the Present Invention

All pending claims stand finally rejected under 35 U.S.C. § 103, as being unpatentable over the *Brock et al.* patent. As a general overview, it is to be noted that the *Brock* patent is directed to an entirely different objective from the present invention. As a result, it does not suggest any solution to the problem addressed by the present invention. More particularly, the *Brock* patent is concerned with the amount of data that needs to be stored for fonts. As described in the paragraph bridging columns 1 and 2, an exemplary font might require about 45 KB of memory, so that a document containing 4 fonts will require about 180 KB just to represent the font information. To alleviate these memory requirements, the *Brock* patent teaches an approach in which the data for each font is not stored in its entirety. Rather, one font, a base or generic font, contains all of the data necessary to generate that font. The other fonts are stored in terms of their differences from the base font. For instance, in the example illustrated in Figures 4-8, the base font represented in Figure 4 is a san-serif font, and another font represented in Figure 5 contains serifs. The tables of Figures 7 and 8 provide the information necessary to generate the serif font of Figure 5 from the san-serif base font of Figure 4. By means of this approach, a library of 100 fonts, which might typically require about 4.5 MB of memory, can be reduced to about 500 KB (column 2, line 63 ff.).

In contrast, the present invention is concerned with the fact that some of the tables that may be needed to properly display a font in accordance with certain line layout technologies may not be present, particularly in the case of older fonts. The *Brock* patent does not address this problem, and more importantly does not disclose the features of the present invention for automatically synthesizing missing font tables. For instance, beginning at column 10, line 24, the *Brock* patent discloses that a base font is implemented

as a set of tables. It does not disclose, however, that if one or more required tables is missing, it can be automatically synthesized from data in the other tables. Rather, the *Brock* patent assumes that all of the necessary tables for generating a font will be present.

The final Office Action states that the *Brock* patent discloses the automatic synthesis of a data table, with reference to the procedure for blending fonts that is described in column 12 and illustrated in Figures 30-34. However, the procedure described in this portion of the *Brock* patent is not the same as the present invention. In general, the blending of fonts as described in the *Brock* patent results in the creation of a *new* font that exhibits design features of two or more available fonts. In contrast, the present invention creates a data table that is absent from an *existing* font. Thus, while the *Brock* patent discloses a technique for creating additional, new fonts, the process of the present invention adds to the functionality of existing fonts, by synthesizing a table for those fonts that enables them to be employed with newer font layout techniques, and the like.

Since the *Brock* patent is concerned with a problem that is wholly unrelated to that addressed by the present invention, it does not disclose or suggest the claimed features of the present invention, as discussed below.

B. The Rejection Does Not Establish a *Prima Facie* Case of Obviousness

The Manual of Patent Examining Procedure, at Section 2143, sets forth three basic criteria for a *prima facie* case of obviousness. At issue on this appeal is the third of those criteria, namely "[t]o establish *prima facie* obviousness of a claimed invention all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)." MPEP §2143.03 (emphasis added). The final rejection merely discusses the claim limitations in general terms, with broad, vague references to the *Brock* patent. It fails to identify how the reference teaches or suggests a number of the specific features of the claims.

The differences between the present invention and the *Brock* patent become apparent upon a detailed analysis of representative claims.

1. Claims 1, 11, 19 and 26

Turning to representative claim 1, the first claimed step recites "retrieving glyphs from a font which correspond to characters in a string of characters." At column 10, lines 24-42, the *Brock* patent discloses that the base font 12 is implemented as a set of tables which contain data for individual glyphs. For instance, the table Glyph Data, illustrated in Figure 25, contains the contour data for all of the glyphs in the base font (column 11, lines 13-15). Although the *Brock* patent does not discuss the details of a printing or imaging process, it is apparent that this glyph data is retrieved from the base font to generate the images of characters.

The next step recited in claim 1 is that of "determining whether the font contains a predetermined data table that pertains to the layout of glyphs." In connection with this claimed subject matter, the first Office Action referred to the *Brock* patent at column 10, lines 53-67; column 11; and Figures 21-26. The Action stated, "The base font in *Brock* contains a predetermined data table that pertains to the layout of glyphs." The Office Action did not identify, however, which data table is being referred to, and the *Brock* patent does not state that any of the data tables pertains to the "layout" of glyphs, e.g. positioning them relative to one another to form a line of characters.

In any event, whether or not any of the tables disclosed in the *Brock* patent pertains to the layouts of glyphs, the referenced portion of the patent does not disclose, nor otherwise suggest, the step of *determining* whether the font contains a predetermined data table. Rather, the procedure of the *Brock* patent operates on the basis that all necessary tables are present. This is confirmed by the above-quoted statement from the first Office Action, i.e. the base font "contains" a predetermined table. As such, there is no need to *determine* if a particular table is present.

The significance of this determination is brought out in the next step recited in claim 1, namely "automatically synthesizing said data table, based upon data contained in the font, *if* the font is determined not to contain said data table." Thus, in the context of claim 1, the automatic synthesis of a data table is conditional upon its presence or absence in a font. A determination is first made *whether* the font contains a predetermined data

table. Then, *if* the condition should occur that the table is not present, it is automatically synthesized from data contained in the font. Thereafter, the glyphs are laid out in a line, in accordance with the data in the automatically synthesized table, and an image of the laid-out line of glyphs is generated.

The procedure recited in claim 1 operates to create a table for a font *if* that table is not already present in the font. The *Brock* patent does not disclose any such conditionality based upon the presence or absence of a table in a font. In particular, there is no teaching to suggest the step of determining whether the base font contains a predetermined data table. Rather, because it is concerned with a totally different type of operation, the *Brock* patent operates on the assumption that all necessary tables are present.

Furthermore, the patent does not disclose any procedure for automatically synthesizing a missing data table, if one were to be absent. With respect to this point, the final Office Action states, "Brock teaches synthesizing the data table when he discloses the blending of characters or glyphs or fonts in a pseudo-code form in figure 29." First, it is to be noted that the algorithm depicted in Figure 29 does not relate to the blending of characters, glyphs or fonts, as alleged in the Action. Rather, the algorithm depicts the procedure carried out in the generator 16 of Figure 1 to generate characters of a given font from the base font 12 and a descriptor file 14. See column 12, lines 6-7.

Second, the *Brock* patent does not teach or suggest that the algorithm functions to produce a *data table*. The output of the algorithm is a character program for the given font. See column 5, lines 22-24. Each character program functions to generate an image such as those shown in Figures 18-20 of the patent. Nowhere does the patent suggest that the algorithm or the generator produces a data table that is missing from a font.

A significant distinction here is the fact that the procedure of the present invention can be carried out within the context of a single font. For instance, claim 1 recites the steps of "retrieving glyphs from *a* font ...", "determining whether *the* font contains a predetermined data table ...", and "automatically synthesizing *said* data table, based upon data contained in *the* font ..." In other words, information from a font that is being employed to generate images is used for the creation of a data table missing from *that* font.

In contrast, the *Brock* patent does not disclose the generation of a data table that is missing from a font. Rather, the patent discloses procedures for utilizing *one* font to generate *another* new font. Thus, while the *Brock* patent discloses a technique for automatically generating font information, the result is significantly different from the data table that is constructed in accordance with the present invention. In the present invention, a new set of data is constructed for an existing font. In the *Brock et al.* patent, a new font is generated from one or more pre-existing fonts.

These same distinctions apply to claims 11, 19 and 26, as well as their dependent claims. Claim 11 recites that the font table synthesizer is "responsive to the absence of a predetermined data table." Claims 19 and 26 recite that the data table is synthesized "if the table is not present in the font file." These features are not taught in the *Brock* patent.

2. Claims 6, 16, 22 and 29

As another point of distinction, the particular manner in which a data table is constructed in a preferred embodiment of the present invention is significantly different from the process for generating a new font in the technique of the *Brock* patent. For example, claim 6 recites the steps of "building a font map that contains information about individual glyphs in the font"; "determining relationships between items of information in the font map"; and "constructing a table which identifies said relationships." Similar recitations are found in claims 16, 22 and 29. With respect to this subject matter, the first Office Action alleged that the *Brock* patent discloses a font map at column 6, lines 28-67; column 7, lines 1-29; and column 11, lines 13-49. It is not apparent what information disclosed in these portions of the patent is considered to constitute a font map.

The Advisory Action dated February 15, 2002 states "In figures 4-8, Brock teaches building a font map (fig. 4-5), determining relationships (fig. 7-8) and constructing a table (fig. 6) . . ." Again, the manner in which the *Brock* patent is being interpreted to disclose these features is not apparent, nor explained. For instance, Figure 4 is described as showing the stem of an exemplary glyph, and Figure 5 shows how the stem of Figure 4 is modified to produce a serifed stem. (Column 3, lines 19-23 and column 6, lines 15-17 and

52-54). These exemplary images of a portion of glyphs do not constitute a font map, such as that illustrated in Figure 7 of the present application. They merely serve to illustrate a concept, and do not constitute a data structure from which a new table is synthesized.

Furthermore, the images of Figures 4 and 5 do not provide information about individual glyphs in "the" font, as recited in the claims. Rather, they illustrate the differences between respective glyphs in two *different* fonts. The table of Figure 6 comprises a quantification of those differences. In other words, it describes relationships between different fonts, not the relationship of items of information for a given font.

3. Claims 7 and 8

Claims 7 and 8 recite further features of the invention relating to the manner in which the font table is synthesized. Claim 7 recites that some of the information in the font map is specific to the font, whereas other information is generic to multiple fonts. Claim 8 recites that the synthesized table contains font-specific information that is determined with reference to generic information. Although both of these claims have been rejected, the Office Actions do not provide any support in the *Brock* patent for their rejection. Without such a showing, the rejection is not supportable.

4. Claims 10, 15, 21 and 28

Claim 10 recites that a determination is made whether a missing data table is one of two different types, and the table is directly synthesized if it is of one type. However, if it is of the other type, an indication is provided that the table is not present, and request is awaited before the synthesizing step is initiated. Similar recitations appear in claims 21 and 28. Claim 15 also recites the two different modes of operation. In rejecting claims 10, 21 and 28, the first Office Action broadly referred to columns 5-10 and Figures 9-23 of the *Brock* patent. However, the Office Action did not explain how any of these portions of the patent can be interpreted to disclose that different approaches for synthesizing a data table are employed in accordance with the type of table that is missing from a font. The *Brock* patent simply does not disclose such a concept.

5. Claims 4, 5, 13, 14, 20 and 27

Claim 4 recites that the synthesized data table is stored in a persistent annex file that is associated with, but separate from, the font. Claim 5 recites that a determination is made whether a missing data table is stored in an annex file. The rejection of these claims analogizes the annex file to the descriptor files 14 of the Brock patent.

The descriptor file of the Brock patent contains specifications for operating upon a base font to produce a desired output font. See, e.g. the Abstract. An example of a descriptor file is shown in Figure 7 of the patent (column 7, lines 3-5). These files are not the same as the annex file recited in the claims. Each descriptor file of the Brock patent pertains to a respectively *different* font. It contains the information needed to generate its particular font. In contrast, the annex file of the present invention contains *additional* synthesized data tables beyond those which form the original font definition. The two types of files serve entirely different purposes, and are used in substantially different manners.

Furthermore, with respect to claims 5 and 14, there is no disclosure in the Brock patent that a determination is made whether a data table is stored in an annex file as part of the process for determining whether the data table needs to be synthesized. The system of the Brock patent doesn't check the descriptor files to see if they contain a particular data file.

IX. Conclusion

From the foregoing, it can be seen that the *Brock* patent is directed to a significantly different objective from the present invention, and as a result does not suggest the present invention to one of ordinary skill in the art. The final rejection fails to identify how the *Brock* patent could be interpreted to disclose, or otherwise suggest, a number of the claimed elements. Consequently, the Examiner has failed to establish a *prima facie* case of obviousness, as required for a rejection under 35 U.S.C. § 103.

The rejection of the claims is not properly founded in the statute, and should be reversed.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: 

James A. LaBarre

Registration No. 28,632

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

Date: May 22, 2002

APPENDIX A

The Appealed Claims

1. A method for generating an image of a sequence of characters, comprising the steps of:

retrieving glyphs from a font which correspond to characters in a string of characters;

determining whether the font contains a predetermined data table that pertains to the layout of glyphs;

automatically synthesizing said data table, based upon data contained in the font, if the font is determined not to contain said data table;

laying out the glyphs in a line, in accordance with the data in said table; and generating an image of the laid-out line of glyphs.

2. The method of claim 1 wherein said generating step includes displaying the line of glyphs on a display device.

3. The method of claim 1 wherein said generating step includes printing the line of glyphs in a document.

4. The method of claim 1 further including the step of storing the synthesized data table in a persistent annex file that is associated with, but separate from, the font.

5. The method of claim 1 further including the step of determining whether said data table is stored in an annex file associated with the font, and wherein said automatic synthesis step is carried out only if the table is not contained in either the font or the annex file.

6. The method of claim 1 wherein the step of automatically synthesizing said data table comprises the steps of:

building a font map that contains information about individual glyphs in the font;

determining relationships between items of information in the font map; and constructing a table which identifies said relationships.

7. The method of claim 6 wherein some of the information in said font map is specific to the font, and other information is generic to multiple fonts.

8. The method of claim 7 wherein the synthesized table contains font-specific information that is determined with reference to generic information.

9. The method of claim 1 wherein the step of automatically synthesizing said data table comprises the steps of retrieving data from the font and storing the retrieved data in a table having a predetermined data format.

10. The method of claim 1 further including the steps of determining whether said data table is of a first type or a second type when the data table is determined not to be present in the font; directly initiating said synthesizing step if said data table is of said first type; or, providing an indication that said data table is not present in the font if said data table is of said second type, and initiating said synthesizing step upon receipt of a request that is responsive to said indication.

11. A system for generating images of characters, comprising:
 - a font subsystem which is responsive to identification of characters to access at least one font file to retrieve glyphs associated with the identified characters, and data tables that contain information about glyphs in the font; and
 - a font table synthesizer which is responsive to the absence of a predetermined data table for creating and storing said table on the basis of data contained in the font file.
12. The system of claim 11 wherein said font subsystem determines whether a predetermined data table is contained in the font file, and causes said synthesizer to create said table when a determination is made that the table is not present in the font file.
13. The system of claim 11 wherein said font synthesizer stores said table in an annex file that is associated with, but separate from, the font file.
14. The system of claim 13 wherein said font subsystem determines whether a predetermined data table is contained in either the font file or the annex file, and causes said synthesizer to create said table when a determination is made that the table is not present in either the font file or the annex file.
15. The system of claim 12 wherein said font subsystem operates in a first mode to cause said synthesizer to automatically create the table in response to said determination, and in a second mode to provide an indication when a data table is determined not to be present and thereafter cause said synthesizer to create the table in response to a request that is responsive to said indication.

16. A method for automatically synthesizing a data table that contains information about glyphs in a font, comprising the steps of:

building a font map that contains information about individual glyphs in the font;

determining relationships between items of information in the font map; and constructing a table which identifies said relationships.

17. The method of claim 16 wherein some of the information in said font map is specific to the font, and other information is generic to multiple fonts.

18. The method of claim 17 wherein the synthesized table contains font-specific information that is determined with reference to generic information.

19. A method for providing data that relates to the implementation of a font, comprising the steps of:

receiving a request for a data table that pertains to the implementation of a font;

determining whether the data table is present in a file containing the font; and

synthesizing said table from data contained in said file if the table is not present in the font file.

20. The method of claim 19 further including the step of storing the synthesized table in an annex file separate from said font file.

21. The method of claim 19 further including the steps of determining whether said data table is of a first type or a second type when the data table is determined not to be present in the font file; automatically initiating said synthesizing step if said data table is of said first type; or, providing an indication that said data table is not present in the font file if said data table is of said second type, and initiating said synthesizing step upon receipt of a request that is responsive to said indication.

22. The method of claim 19 wherein the step of synthesizing said data table comprises the steps of:

building a font map that contains information about individual glyphs in the font;

determining relationships between items of information in the font map; and constructing a table which identifies said relationships.

23. The method of claim 22 wherein some of the information in said font map is specific to the font, and other information is generic to multiple fonts.

24. The method of claim 23 wherein the synthesized table contains font-specific information that is determined with reference to generic information.

25. The method of claim 19 wherein the step of synthesizing said data table comprises the steps of retrieving data from the font and storing the retrieved data in a table having a predetermined data format.

26. A computer-readable medium containing a program which executes the steps of:

receiving a request for a data table that pertains to the implementation of a font;

determining whether the data table is present in a file containing the font; and

synthesizing said table from data contained in said file if the table is not present in the font file.

27. The computer-readable medium of claim 26, wherein said program executes the further step of storing the synthesized table in an annex file separate from said font file.

28. The computer-readable medium of claim 26, wherein said program executes the further step of determining whether said data table is of a first type or a second type when the data table is determined not to be present in the font file; automatically initiating said synthesizing step if said data table is of said first type; or, providing an indication that said data table is not present in the font file if said data table is of said second type, and initiating said synthesizing step upon receipt of a request that is responsive to said indication.

29. A computer-readable medium containing a program which executes the steps of:

building a font map that contains information about individual glyphs in a font;

determining relationships between items of information in the font map; and constructing a table which identifies said relationships.

30. The computer-readable medium of claim 29, wherein some of the information in said font map is specific to the font, and other information is generic to multiple fonts.

31. The computer-readable medium of claim 30, wherein the synthesized table contains font-specific information that is determined with reference to generic information.